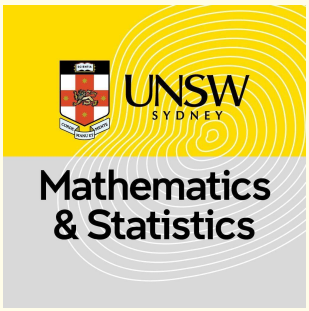


MATH5805
A Modern Introduction to Extreme Value Theory
Term 1, 2025

Dr. Boris Béranger

Week 4 - Lab Solutions

Exercise 1

Suppose that $X_1, \ldots, X_n \sim N(0,1)$. Generate $n = 100,000$ samples of X_i.

a) Consider an increasing sequence of thresholds $u = 0, 0.52, 1.28, 1.64, 1.96, 2.33, 2.58 \ldots$ (these are the 0.5, 0.7, 0.9, 0.95,0.99,0.995 quantiles of the $N(0,1)$), and produce histograms of the threshold exceedances $Y = \{X_i - u | X_i > u\}$. Note the change in location, scale and shape of the distribution as u increases.

Solution:

```
library(extRemes)
set.seed(123)
x <- rnorm(100000)
u <- qnorm(c(0.5,0.7,0.9,0.95,0.99,0.995))
m <- max(x)
par(mfrow=c(3,2))
for (i in 1:6) {
  #fit <- fevd(x,threshold=u[i],type="GP") # This runs more slowly
  fit <- fevd(x[x>u[i]], threshold=u[i], type="GP") # Faster
  p <- fit$results$par
  ymax <- devd(0, scale=p[1], shape=p[2], threshold=0, type="GP")
  xx <- seq(0, m-u[i], length=100)
  yy <-devd(xx, scale=p[1], shape=p[2], threshold=0, type="GP")
  #hist(x[x>u[i]]-u[i], main=paste("u =",round(u[i],3)), probability=T,
  #     xlab="x-u|x>u", ylim=c(0,max(yy)))
  #lines(xx, yy)
}
```

The distribution of the normal tail is noticeably different from the GPD approximation for $u = 0, 0.524$, but seems (to the limit of these plots) to become more acceptable as u becomes larger.

b) For each u fit the GPD distribution to the threshold exceedances. Explore how the distribution of the block maxima approaches the GPD limit as $u \to \infty$ by a) superimposing the fitted density on the histograms b) examining goodness of fit diagnostics (such as qq-plots) and c)

producing profile likelihood confidence intervals for ξ (what value should ξ approach?). How fast does the GPD approximation become a good one?

Solution:

```
for(i in 1:6){
  cat("With threshold u=", u[i], "\n")
  fit <- fevd(x[x>u[i]], threshold=u[i], type="GP")
  # plot(fit)
  ci <- ci(fit,type="parameter")
  cat("Estimate of the shape is ", ci[4], "with ci (", ci[2], ",", ci[6], ")\n")
}
```

```
## With threshold u= 0
## Estimate of the shape is  -0.2212143 with ci ( -0.2212143 , -0.2212142 )
## With threshold u= 0.5244005
## Estimate of the shape is  -0.1943988 with ci ( -0.1979638 , -0.1908338 )
## With threshold u= 1.281552
## Estimate of the shape is  -0.1479547 with ci ( -0.1628784 , -0.1330311 )
## With threshold u= 1.644854
## Estimate of the shape is  -0.1219834 with ci ( -0.1454214 , -0.09854538 )
## With threshold u= 2.326348
## Estimate of the shape is  -0.04732715 with ci ( -0.1102498 , 0.01559556 )
## With threshold u= 2.575829
## Estimate of the shape is  -0.06881191 with ci ( -0.1633893 , 0.02576543 )
```

The goodness-of-fit plots show a clear divergence of the tail behaviour of the $N(0,1)$ from a GPD for $i = 1, 3$ (mostly clearly seen in the return level plot). For $i = 6$ the approximation is starting to look ok. However examination of the 95% CI sfor ξ shows that this parameter is still converging to some value (which we know to be 0 in this case), so clearly there is still some way to go before the asymptotics really kick in. Again, the normal distribution is exhibiting very slow asymptotic behaviour.

c) The normal distribution is in the domain of attraction of the Gumbel distribution, which results in an exponential distribution for the GPD limit. Repeat question b) but fitting the Exponential distribution instead of the GPD. Contrast your results with the GPD fit. Is there an advantage in knowing that the domain of attraction is Gumbel, compared to not knowing this?

Solution:

```
for(i in 1:6){
  cat("With threshold u=", u[i], "\n")
  fit <- fevd(x[x>u[i]], threshold=u[i], type="Exponential")
  # plot(fit)
}
```

```
## With threshold u= 0
## With threshold u= 0.5244005
```

```
## With threshold u= 1.281552
## With threshold u= 1.644854
## With threshold u= 2.326348
## With threshold u= 2.575829
```

It is now very much apparent that even for $i = 6$ ($u = 2.676$, the 99.5th quantile of the standard normal), the limiting exponential distribution tail behaviour has not yet been achieved. Previously when fitting the GPD with the extra shape parameter, this allowed the fitted density to mask the lack of convergence. With our knowledge that the limit should be exponential, the lack of convergence is now very much apparent.

Exercise 2

Repeat questions a) & b) from Exercise 1 but with $X_1, \ldots, X_n \sim U(0, 1)$. Is the GPD convergence faster or slower than for the normal distribution?

Solution:

```
set.seed(123)
x <- runif(100000)
u <- qunif(c(0.5,0.7,0.9,0.95,0.99,0.995))
m <- max(x)
par(mfrow=c(3,2))
for(i in 1:6) {
  fit <- fevd(x[x>u[i]], threshold=u[i], type="GP")
  p <- fit$results$par
  ymax <- devd(0, scale=p[1], shape=p[2], threshold=0, type="GP")
  xx <- seq(0, m-u[i], length=100)
  yy <- devd(xx, scale=p[1], shape=p[2], threshold=0, type="GP")
  # hist(x[x>u[i]]-u[i], main=paste("u =",round(u[i],3)), probability=T,
  #      xlab="x-u|x>u", ylim=c(0,max(yy)))
  # lines(xx, yy)
}
```

All of these fits are qualitatively the same. The limiting tail behaviour of a uniform is clearly a uniform, however the GPD has the capacity to accurately represent this limit. (The mathematics for this was shown in the lectures.) Exploring the goodness-of-fit:

```
for(i in 2:6){
  cat("With threshold u=", u[i], "\n")
  fit <- fevd(x[x>u[i]], threshold=u[i], type="GP")
  # plot(fit)
  ci <- ci(fit,type="parameter")
  cat("Estimate of the shape is ", ci[4], "with ci (", ci[2], ",", ci[6], ")\n")
}
```

```
## With threshold u= 0.7
## Estimate of the shape is  -0.9999878 with ci ( -0.9999878 , -0.9999878 )
## With threshold u= 0.9
```

```
## Estimate of the shape is  -1.005825 with ci ( -1.005825 , -1.005825 )
## With threshold u= 0.95
## Estimate of the shape is  -0.9999283 with ci ( -0.9999283 , -0.9999283 )
## With threshold u= 0.99
## Estimate of the shape is  -0.9571707 with ci ( -0.9571708 , -0.9571707 )
## With threshold u= 0.995
## Estimate of the shape is  -0.9779279 with ci ( -0.9779279 , -0.9779278 )
```

The above goodness-of-fit diagnostic is for $i = 5$, but all threshold fits show essentially the same behaviour (simply with more or less data). As anticipated the fit is almost perfect, and the shape parameter is estimated as $\hat{\xi} = -1.00547592$ as expected. Convergence here is effectively instantaneous for any threshold in stark contrast to the normal distribution.

Exercise 3

Repeat questions a) & b) from Exercise 1 but with $X_1, \ldots, X_n \sim$ log-Gamma$(1, 1)$. Comment on the speed of convergence compared to the normal and the uniform. (Recall that if $Y \sim$ Gamma(a, b) then $X = \exp Y \sim$ log-Gamma(a, b).) You may wish to limit the range of the x-axis on your plots for this heavy-tailed distribution.

Solution:

```
a <- 1
b <- 1
set.seed(123)
x <- sort(exp(rgamma(100000, a, b)))
u <- x[c(0.5,0.7,0.9,0.95,0.99,0.995)*100000] # approximate quantiles
m <- x[1000000] # max value
llim <- c(30, 50, 125, 400, 1125, 5000)
par(mfrow=c(3,2))
for(i in 1:6) {
  fit <- fevd(x[x>u[i]], threshold=u[i], type="GP")
  p <- fit$results$par
  ymax <- devd(0, scale=p[1], shape=p[2], threshold=0, type="GP")
  xx <- seq(0, llim[i], length=100)
  yy <- devd(xx, scale=p[1], shape=p[2], threshold=0,type="GP")
  temp <- x[x>u[i]]-u[i]
  # hist(temp[temp<llim[i]], main=paste("u =",round(u[i],3)), probability=T,
  #      xlab="x-u|x>u", xlim=c(0,llim[i]-u[i]), ylim=c(0,max(yy)), nclass=31)
  # lines(xx,yy)
}
```

```
for(i in 1:6){
  fit <- fevd(x[x>u[i]], threshold=u[i], type="GP")
  #plot(fit)
  ci <- ci(fit,type="parameter")
  cat("Estimate of the shape is ", ci[4], "with ci (", ci[2], ",", ci[6], ")\n")
}
```

```
## Estimate of the shape is  1.000519 with ci ( 0.9829825 , 1.018055 )
## Estimate of the shape is  0.9957916 with ci ( 0.9732933 , 1.01829 )
## Estimate of the shape is  1.023687 with ci ( 0.9840701 , 1.063305 )
## Estimate of the shape is  1.044996 with ci ( 0.9885174 , 1.101474 )
## Estimate of the shape is  1.106589 with ci ( 0.9760606 , 1.237117 )
## Estimate of the shape is  1.03436 with ci ( 0.8581487 , 1.21057 )
```

All of these plots suggest that convergence to the GPD tail behaviour limit is effectively instantaneous. With the 95% CI for ξ being (0.9873343,1.001616) for $i = 2$ we have clearly identified the heavy Fréchet tail behaviour. So both log-Gamma and Uniform tail behaviour appears to be GPD instantaneously, in sharp contrast to the normal.
Note that for the GPD, the shape parameter does not depend on the threshold. However the scale parameter *does*, and so will take different values for different u.

Exercise 4

Consider again the historical AUD/xxx daily exchange rates data from the Week 3 lab, given in the file `AUD-exchangeRates.csv`, and again construct the log daily returns of the AUD/USD (column FXRUSD) series.

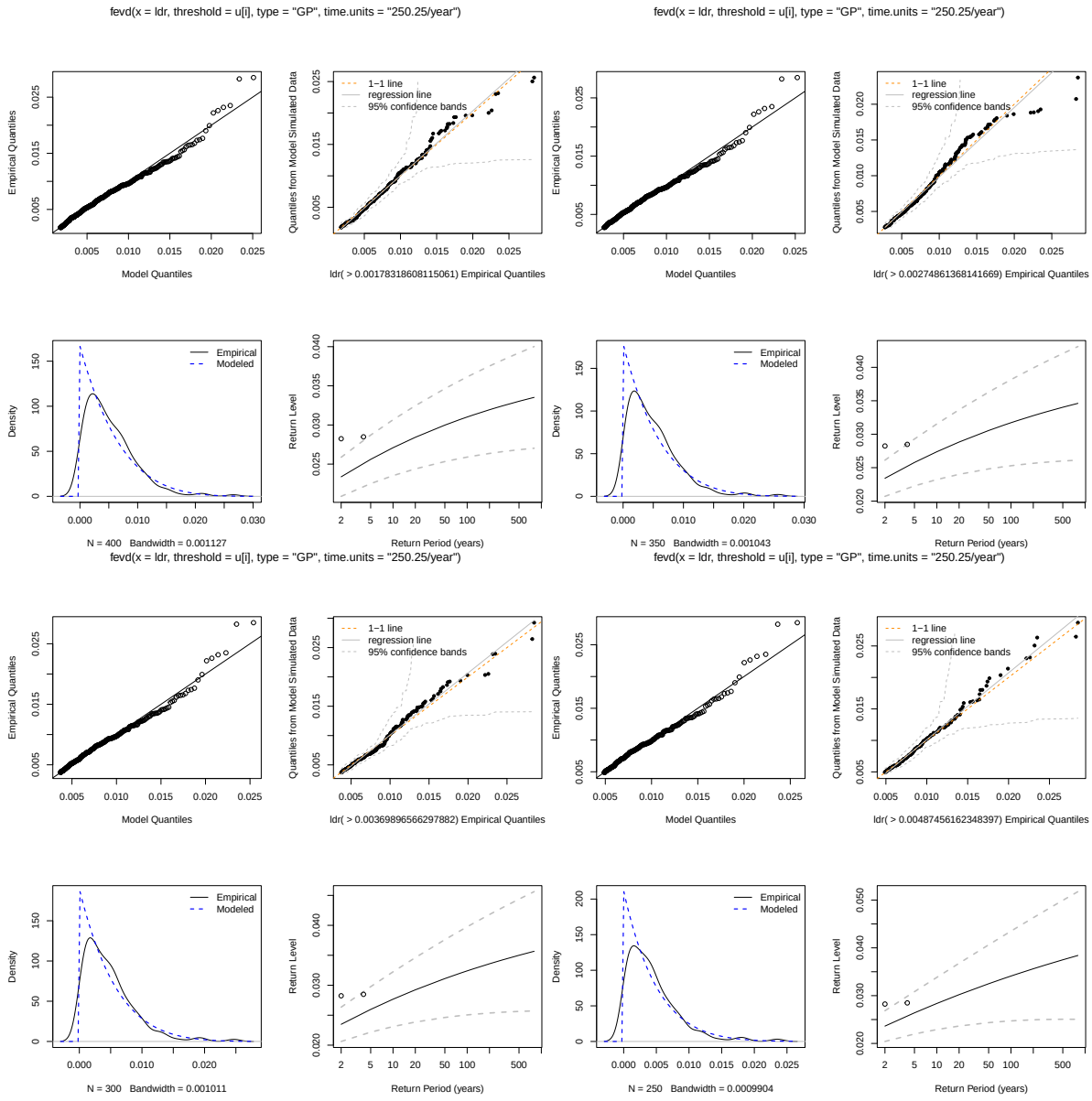
In the Week 3 lab, we explored how the block maxima of this series, taken over increasing block sizes (daily, weekly, monthly, quarterly, annually) converged to the GEV limit, for large enough block sizes. Let's repeat this exercise from the threshold exceedance perspective. We will explore how to choose an appropriate threshold in different ways.

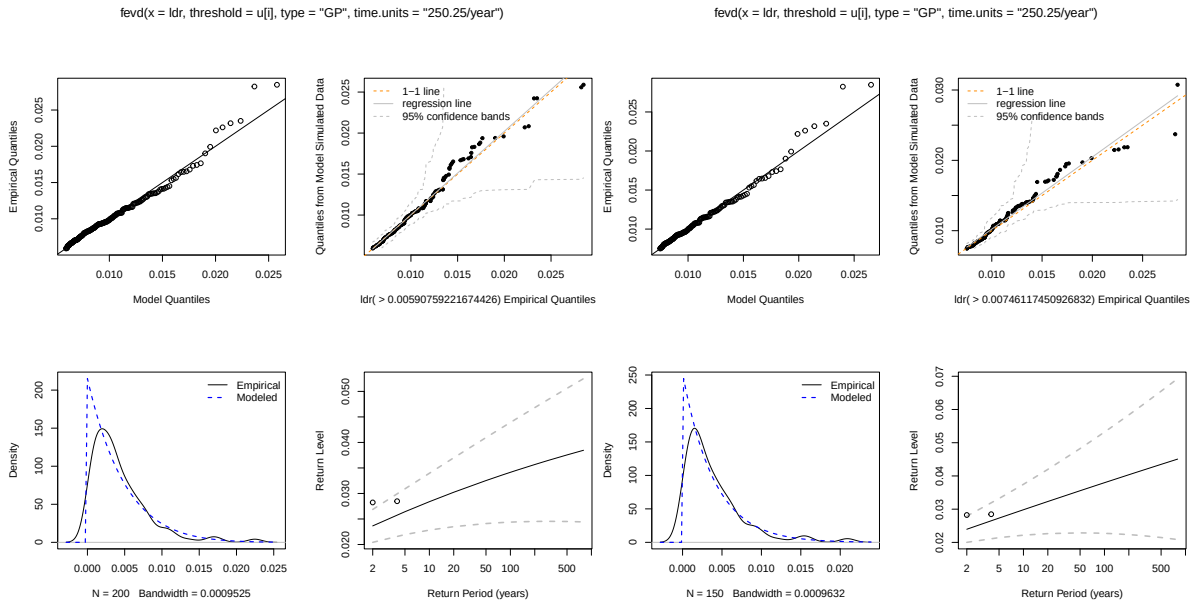
a) For thresholds, u, determined as the 60%, 65%, 70%, ..., 90%, 95% upper quantiles of the log daily returns, fit a GPD to the threshold exceedances and use goodness-of-fit diagnostics to assess the convergence of the distribution's tail behaviour to the GPD limit. Note that in `fevd` you should specify how many observations are in a given time period, so that the return level plots will be on the right scale. E.g.~`time.units="250.25/year"`.

What is the lowest threshold for which the approximation to the GPD limit is acceptable?

Solution:

```
library(extRemes)
dat <- read.table("../data/AUD-exchangeRates.csv", header=TRUE, sep=",")
usaud <- dat$FXRUSD
L <- length(usaud)
ldr <- log(usaud[2:L]/usaud[1:(L-1)])
u <- quantile(ldr,c(.6,.65,.7,.75,.8,.85,.9,.95))
for(i in 1:6){
  fit <- fevd(ldr, threshold=u[i], type="GP", time.units="250.25/year")
  plot(fit)
}
```


fevd(x = ldr, threshold = u[i], type = "GP", time.units = "250.25/year")
Empirical Quantiles
Model Quantiles
Quantiles from Model Simulated Data
ldr(> 0.00178318608115061) Empirical Quantiles
1−1 line
regression line
95% confidence bands
Density
N = 400 Bandwidth = 0.001127
Empirical
Modeled
Return Level
Return Period (years)
fevd(x = ldr, threshold = u[i], type = "GP", time.units = "250.25/year")
ldr(> 0.00274861368141669) Empirical Quantiles
N = 350 Bandwidth = 0.001043
fevd(x = ldr, threshold = u[i], type = "GP", time.units = "250.25/year")
ldr(> 0.00369896566297882) Empirical Quantiles
N = 300 Bandwidth = 0.001011
fevd(x = ldr, threshold = u[i], type = "GP", time.units = "250.25/year")
ldr(> 0.00487456162348397) Empirical Quantiles
N = 250 Bandwidth = 0.0009904



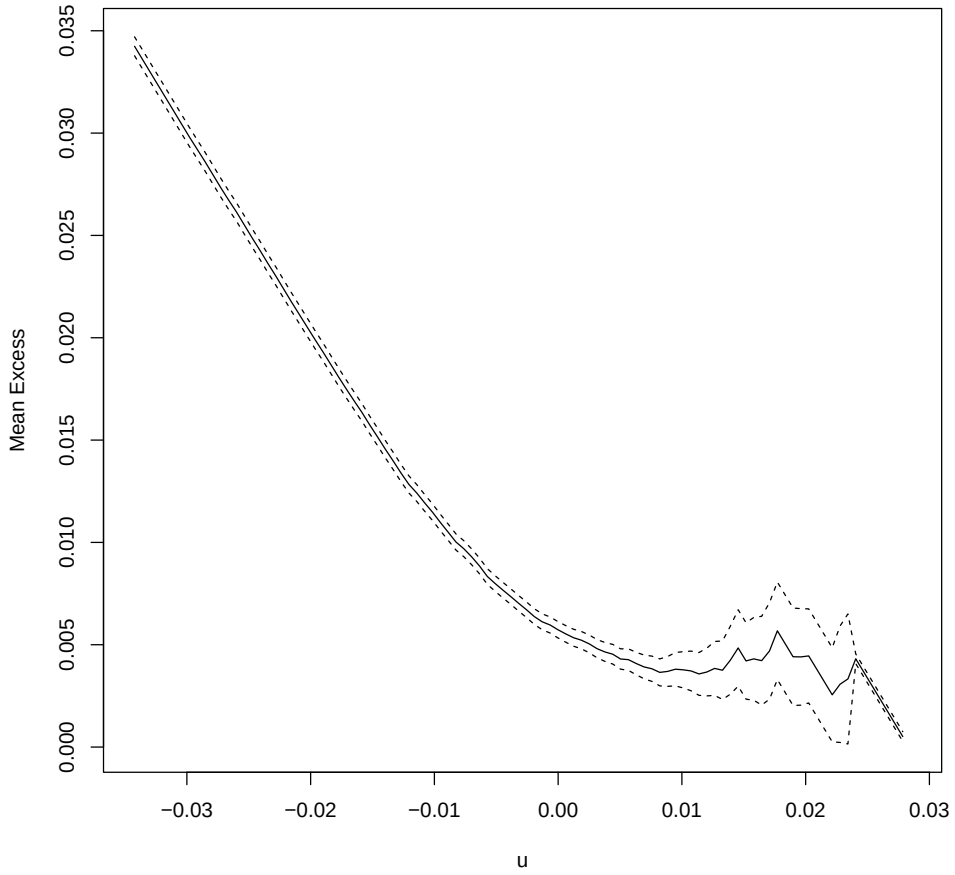
For a threshold at the 70% quantile, the qq-plot (top left panel) shows that the data are systematically under the line $y = x$ for a good proportion of the tail, indicating a lack of fit. This feature is still there at for a threshold at the 80% quantile, but has noticeably disappeared for the 85% quantile ($u = 0.00746$). For thresholds above this, the fit is also acceptable. Therefore we adopt $u = 0.00746$ as our threshold choice (to the nearest 5% empirical quantile).

b) Produce both a mean excess plot (mean residual life plot) and a plot of GPD parameter value versus threshold. Does these plots support your previous choice of threshold?

(Note: the `extRemes` function `mrlplot` does not allow the user to specify the range of threshold values to be considered, and instead fixes this range as (min(data), max(data)-1). Given the scale of our data, the hard coded -1 is not helpful! As an alternative, use the `mrl.plot` function from the `ismev` package, which does allow the threshold range to be specified, via `umin` and `umax`.)

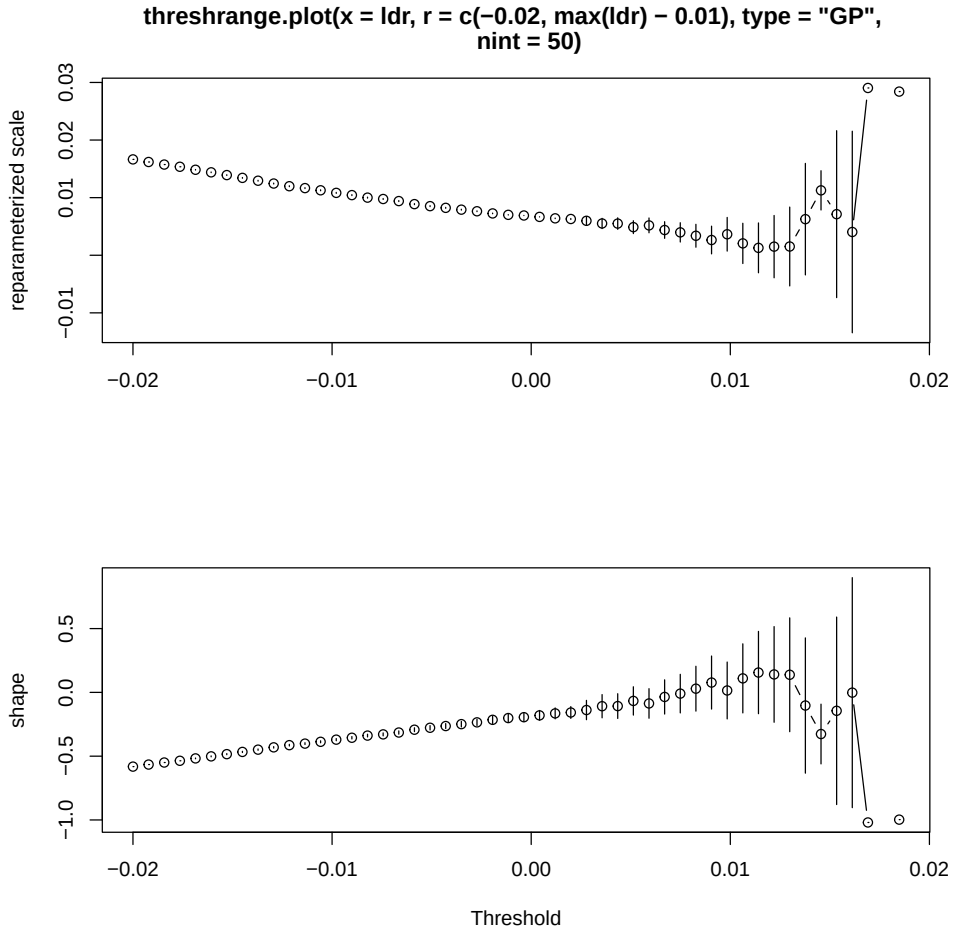
Solution:

```
library(ismev)
mrl.plot(ldr, umin=min(ldr), umax=max(ldr))
```




```
threshrange.plot(ldr, r=c(-0.02,max(ldr)-0.01), nint=50, type="GP")
```

```
## Error in get(paste("ci.", newcl, sep = ""))(x = x, alpha = alpha, type = type,  :
##   ci: Sorry, unable to calculate the parameter covariance matrix.  Maybe try a differer
```



While linear for negative values, the mean excess plot changes direction around zero. Within the bounds of the confidence intervals, the plot could be linear from around $u = 0.007$ or so. The plot of gpd parameter values is clearly changing linearly with u until a little before $u = 0.01$ from which (taking the CIs into account) the parameter estimates could be constant with increasing u.
Taken together, these diagnostics support the previous threshold choice of $u = 0.007$.

c) As the parameters of the GPD$(\tilde{\sigma}, \xi)$ can be related to those of the GEV(μ, σ, ξ) through $\tilde{\sigma} = \sigma + \xi(u - \mu)$, show that your GPD parameter estimates are consistent with the GEV estimates (with your chosen block size, n) that you obtained in your block maxima analysis of these data in Lab 2.

Solution:

```
nn <- 21# approximate no. of data per block
temp <- matrix(ldr,ncol=nn,byrow=T)
temp <- temp[-dim(temp)[1],] # need to remove last row
bm <- apply(as.matrix(temp), 1, max)
fit.gev <- fevd(bm, type="GEV")
fit.gev

##
```

```
## fevd(x = bm, type = "GEV")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  -182.2802
##
##
##  Estimated parameters:
##    location       scale       shape
## 0.010591602 0.004219845 0.017145398
##
##  Standard Error Estimates:
##     location        scale        shape
## 0.0007006716 0.0004407288 0.1129417428
##
##  Estimated parameter covariance matrix.
##               location         scale         shape
## location  4.909407e-07  1.290383e-07 -2.901252e-05
## scale     1.290383e-07  1.942418e-07 -1.248940e-05
## shape    -2.901252e-05 -1.248940e-05  1.275584e-02
##
##  AIC = -358.5604
##
##  BIC = -353.01
```

```
fit.gpd <- fevd(ldr, threshold=u[6], type="GP", time.units="25/month")
fit.gpd
```

```
##
## fevd(x = ldr, threshold = u[6], type = "GP", time.units = "25/month")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  -682.3844
##
##
##  Estimated parameters:
##        scale        shape
##  0.003952107 -0.015727328
##
##  Standard Error Estimates:
##        scale        shape
## 0.0003516057 0.0764504420
##
##  Estimated parameter covariance matrix.
##               scale         shape
```

```
## scale  1.236265e-07 -1.855835e-05
## shape -1.855835e-05  5.844670e-03
##
##  AIC = -1360.769
##
##  BIC = -1354.747
```

```
p.gev <- fit.gev$results$par
p.gpd <- fit.gpd$results$par
p.gev[2]+p.gev[3]*(u[6]-p.gev[1])  # approx equal to GPD scale
```

```
##       scale
## 0.004166173
```

```
ci(fit.gev, type="parameter", which.par=3, method="proflik")
```

```
## fevd(x = bm, type = "GEV")
##
## [1] "Profile Likelihood"
##
## [1] "shape: 0.017"
##
## [1] "95% Confidence Interval: (-0.0087, 0.2742)"
```

```
ci(fit.gpd, type="parameter", which.par=2, method="proflik")
```

```
## fevd(x = ldr, threshold = u[6], type = "GP", time.units = "25/month")
##
## [1] "Profile Likelihood"
##
## [1] "shape: -0.016"
##
## [1] "95% Confidence Interval: (-0.0808, 0.107)"
```

The GEV parameter estimates are $(\hat{\mu}, \hat{\sigma}, \hat{\xi}_{GEV}) = (0.010591602, 0.004219845, 0.017145398)$. With a threshold of $u = 0.007461175$ this should yield GPD parameter estimates of $(\hat{\tilde{\sigma}}, \hat{\xi}_{GPD}) \approx (\hat{\sigma} + \hat{\xi}_{GEV}(u - \hat{\mu}), \hat{\xi}_{GEV}) \approx (0.004166172, 0.017145398)$. The actual GPD parameter estimates are $(0.003952207, -0.015748807)$. The scale parameter matches quite closely, though there is some difference in the shape parameters. Looking at their profile likelihood confidence intervals we have (-0.0087, 0.2742) for the GEV and (-0.081, 0.1069) for the GPD. Although there is some difference between the two, there is considerable overlap allowing (loosely) for the possibility that the parameters are the same.

d) In principle, as the GPD parameter estimates are obtained using more data than the block maxima approach, these should be more precise than the equivalent GEV parameters. Is this the case?

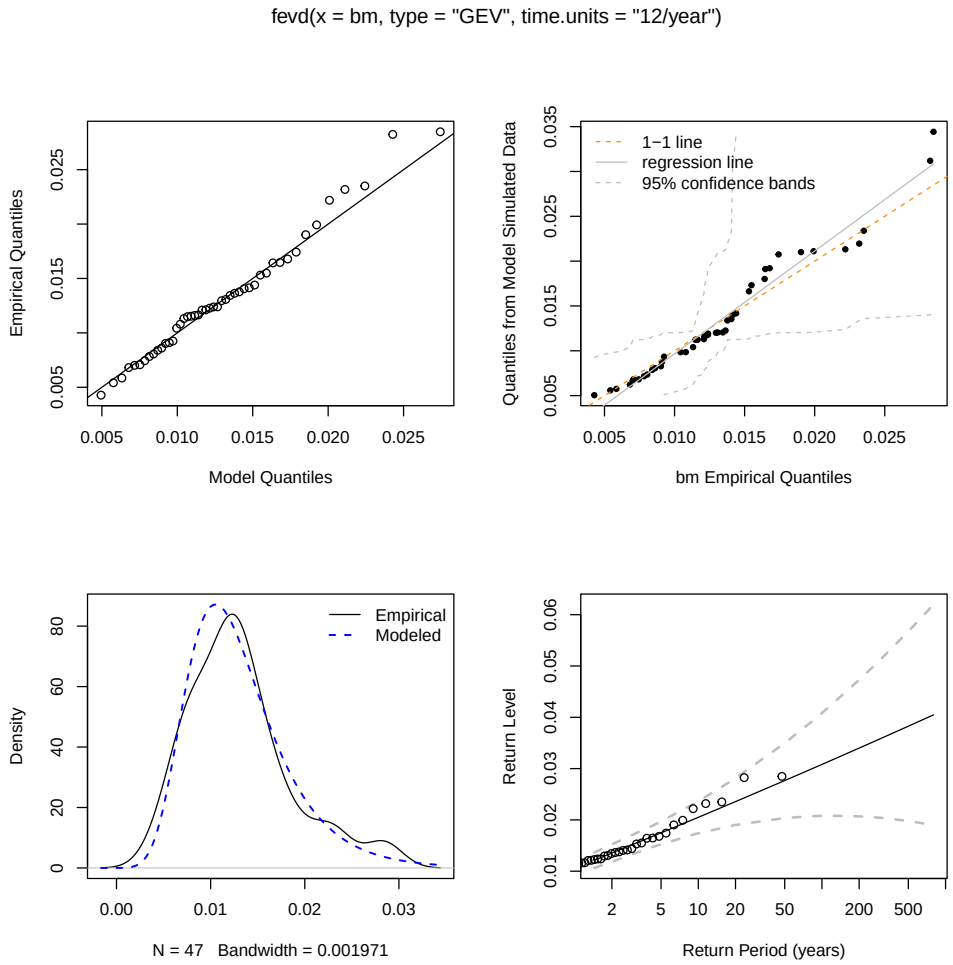
Solution:
The only directly and easily comparable model parameter between the GEV and GPD fits is ξ. From the analysis in question c) the 95% CIs for this parameter are (-0.0087, 0.2742) for the GEV and (-0.081, 0.1069) for the GPD. The latter interval is much shorter than the

former, supporting the idea that the threshold exceedance approach is more data efficient.

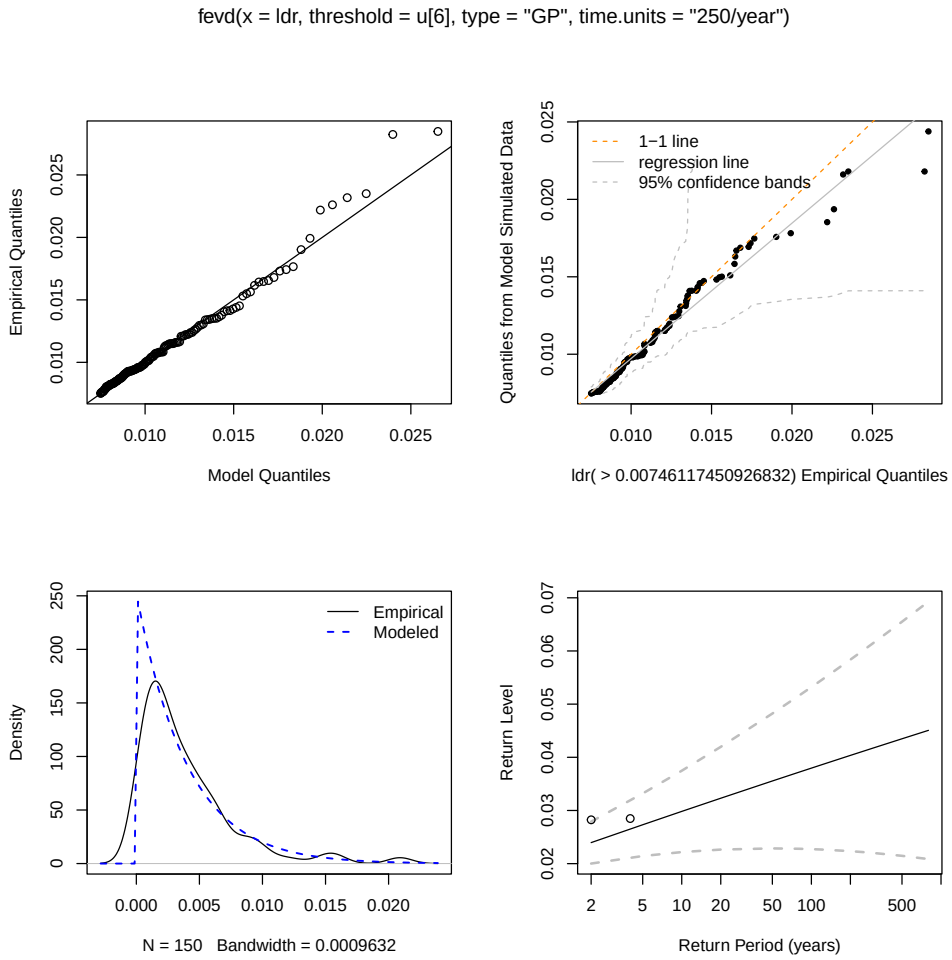
e) Similarly compare return level plots (on the same temporal scale) for the GEV and GPD analyses. Are they consistent? Are the GPD analyses more precise?

Solution:

```
fit.gev=fevd(bm,type="GEV",time.units="12/year")
# time.unit has no effect for GEV
plot(fit.gev)
```




```
fit.gpd <- fevd(ldr, threshold=u[6], type="GP", time.units="250/year")
plot(fit.gpd)
```



When fitting the two models, note that they are on different scales. The GEV is (with $n = 21$) on a monthly scale, and the GPD (once we specify `time.units="250/year"`) is on an annual scale. Changing the time units on the GEV has no effect in these R functions, so it is simpler to compare any GPD return level with the GEV return level at $12\times$ (monthly) the scale. So a 2-year return level for the GPD would be a 24 "year" (i.e. month!) return level for the GEV. With this we obtain

```
return.level(fit.gev, return.period=12*c(2,20,100), do.ci=T)
```

```
## fevd(x = bm, type = "GEV", time.units = "12/year")
##
## [1] "Normal Approx."
##
##                          95% lower CI   Estimate 95% upper CI
## 24-year return level      0.01932050 0.02428012   0.02923975
## 240-year return level     0.02058847 0.03483084   0.04907321
## 1200-year return level    0.01808228 0.04240314   0.06672399
```

```
return.level(fit.gpd, do.ci=T)
```

```
## fevd(x = ldr, threshold = u[6], type = "GP", time.units = "250/year")
##
```

```
## [1] "Normal Approx."
##
##                         95% lower CI   Estimate 95% upper CI
## 2-year return level      0.02001370 0.02395424   0.02789477
## 20-year return level     0.02264407 0.03230491   0.04196575
## 100-year return level    0.02272878 0.03796480   0.05320081
```

As is apparent, and expected, the confidence intervals for the equivalent return levels are smaller for the GPD compared to the GEV.

Exercise 5

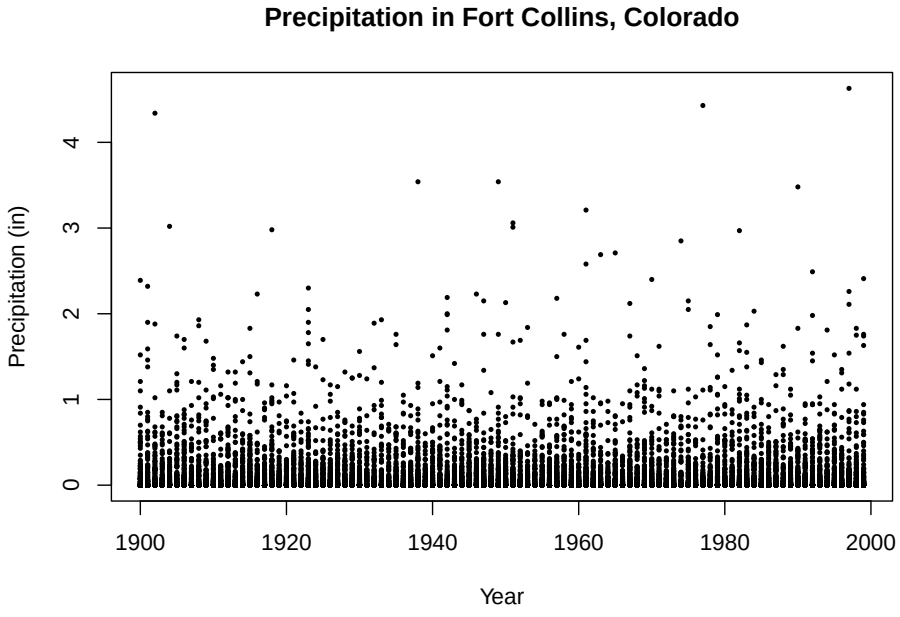
Consider the daily precipitation data at Fort Collins, Colorado, considered in the lectures. This is available in the `extRemes` R package via `data(Fort)`.

a) Briefly repeat the analysis in the lectures:

i) Plot the precipitation data over time – do there appear to be any trends?;

Solution:

```
data(Fort)
plot(Fort$year,Fort$Prec,pch=16,cex=0.5,xlab="Year",
     ylab="Precipitation (in)",
     main="Precipitation in Fort Collins, Colorado")
```



ii) fix a threshold of $u = 0.5$ and fit the GPD to the threshold exceedances;

Solution:

```
u <- 0.5
fit <- fevd(Fort$Prec,threshold=u,type="GP",time.units="365.25/year")
```

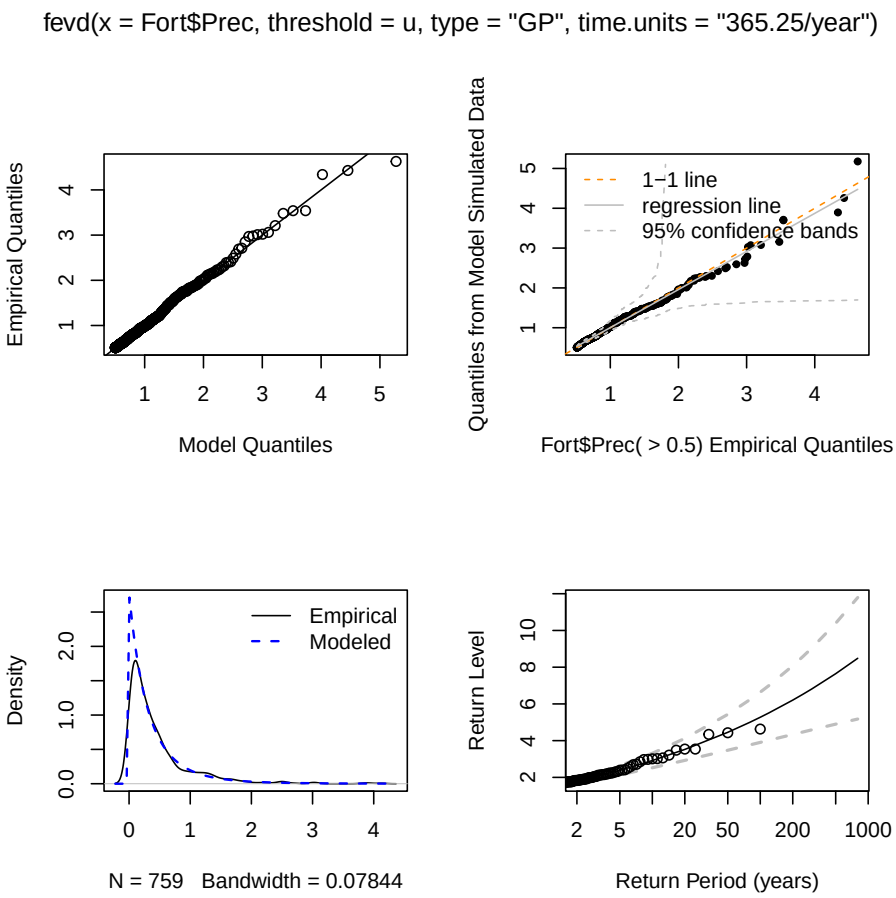
```
fit
```

```
##
## fevd(x = Fort$Prec, threshold = u, type = "GP", time.units = "365.25/year")
##
## [1] "Estimation Method used: MLE"
##
##
##  Negative Log-Likelihood Value:  128.864
##
##
##  Estimated parameters:
##     scale     shape
## 0.3610066 0.1886400
##
##  Standard Error Estimates:
##      scale      shape
## 0.02059776 0.04455210
##
##  Estimated parameter covariance matrix.
##                scale         shape
## scale  0.0004242675 -0.0006099035
## shape -0.0006099035  0.0019848896
##
##  AIC = 261.7279
##
##  BIC = 270.9919
```

iii) assess the fit via goodness-of-fit checks;

Solution:

```
plot(fit)
```



iv) estimate the 2-year, 20-year and 100-year return periods (with confidence intervals).

Solution:

```
return.level(fit, period=c(2,20,100), do.ci=T)
```

```
## fevd(x = Fort$Prec, threshold = u, type = "GP", time.units = "365.25/year")
##
## [1] "Normal Approx."
##
##                        95% lower CI Estimate 95% upper CI
## 2-year return level        1.654159 1.783091     1.912023
## 20-year return level       2.924781 3.522081     4.119380
## 100-year return level      3.905295 5.272977     6.640659
```

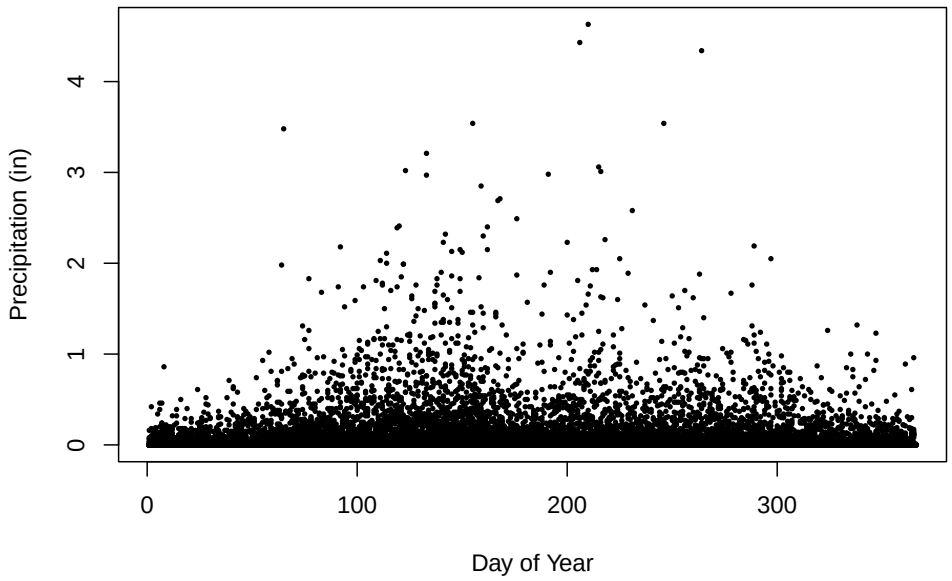
At first glance the data seem stationary, and the model fit a good one.

b) When working with block-maxima representations of extremes, any seasonality that occurs within the block is already accounted for when computing the block maximum. E.g.~Computing the annual maxima temperature is likely to select a temperature that occurs in summer, rather than winter. However, when modelling using threshold-exceedances we need to consider that the distribution of extremes might vary due to seasonality in addition to any changes via longer-term trends. E.g.~A temperature of (say) $35°C$ in winter is more extreme than the same temperature in summer.

Produce a plot of the Fort Collins precipitation data against time of year. Does there seem to be any within-year seasonality present? If so, we should be able to achieve a more precise analysis if we account for this seasonality.

Solution:

```
plot(Fort$tobs,Fort$Prec,pch=16,cex=0.5,xlab="Day of Year",
     ylab="Precipitation (in)")
```



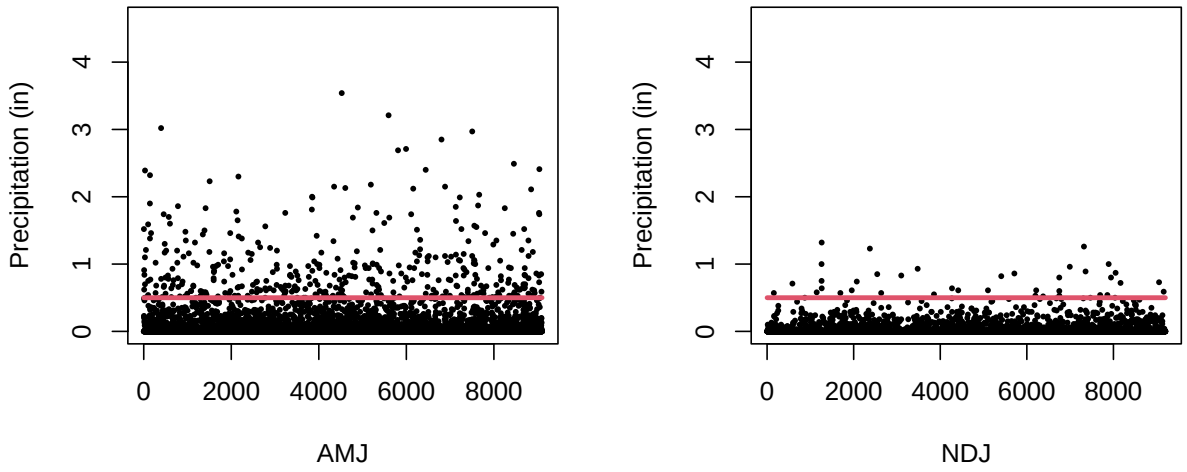
There appears to be a marked increase in rainfall extremes towards the middle of each year compared to at the ends.

c) Consider separately the precipitation data in April-May-June and November-December-January, and fit separate GPD models to this data. (It is not uncommon to account for potential within-year seasonality by splitting up the data in this way.) Check the models' goodness of fit, and produce return level plots and estimate the 2-year, 20-year and 100-year return periods. Are there differences in the extremes at different times of the year?

Solution:

We first plot the respective datasets, including the threshold (horizontal red line).

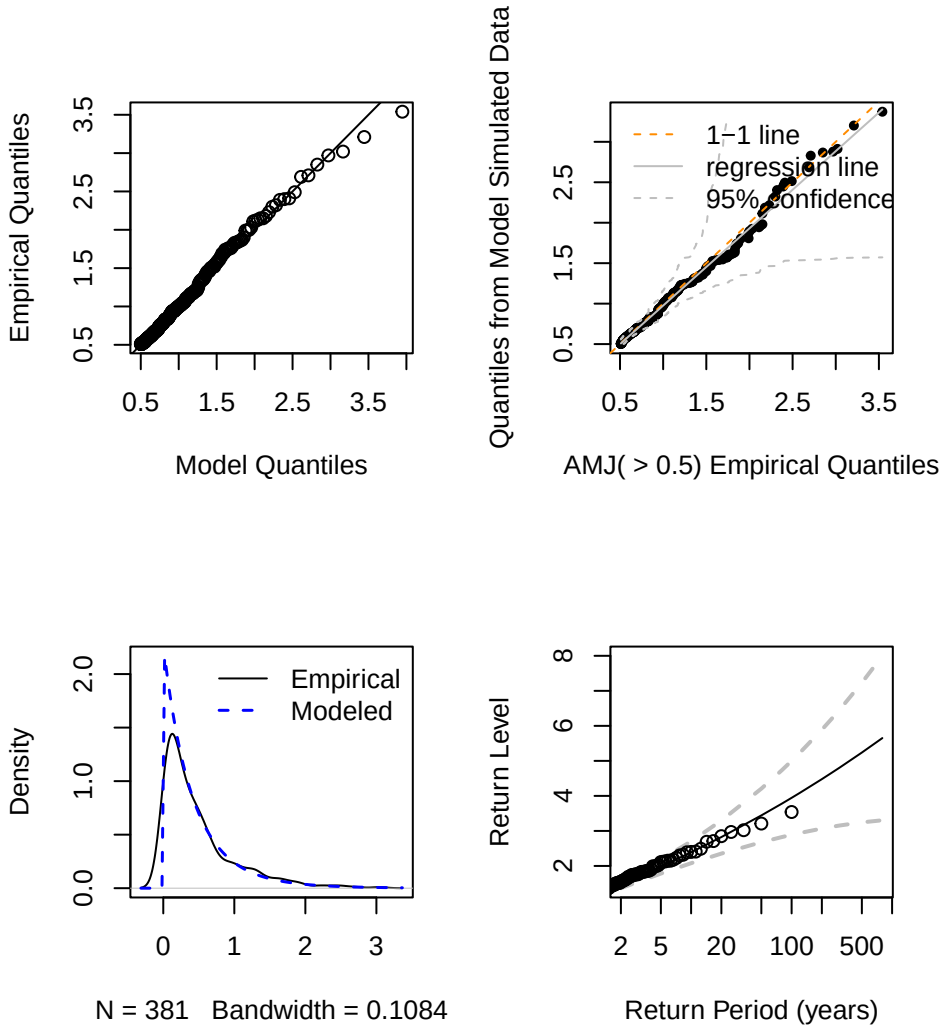
```
AMJ <- Fort$Prec[Fort$month==4 | Fort$month==5 | Fort$month==6]
NDJ <- Fort$Prec[Fort$month==1 | Fort$month==11 | Fort$month==12]
par(mfrow=c(1,2))
r <- range(Fort$Prec)
plot(AMJ, pch=16, cex=0.5, xlab="AMJ", ylab="Precipitation (in)", ylim=r)
lines(c(1,length(AMJ)), c(u,u), col=2, lwd=3)
plot(NDJ, pch=16, cex=0.5, xlab="NDJ", ylab="Precipitation (in)", ylim=r)
lines(c(1,length(NDJ)), c(u,u), col=2, lwd=3)
```



We then fit the GPD models and produce the diagnostic plots.

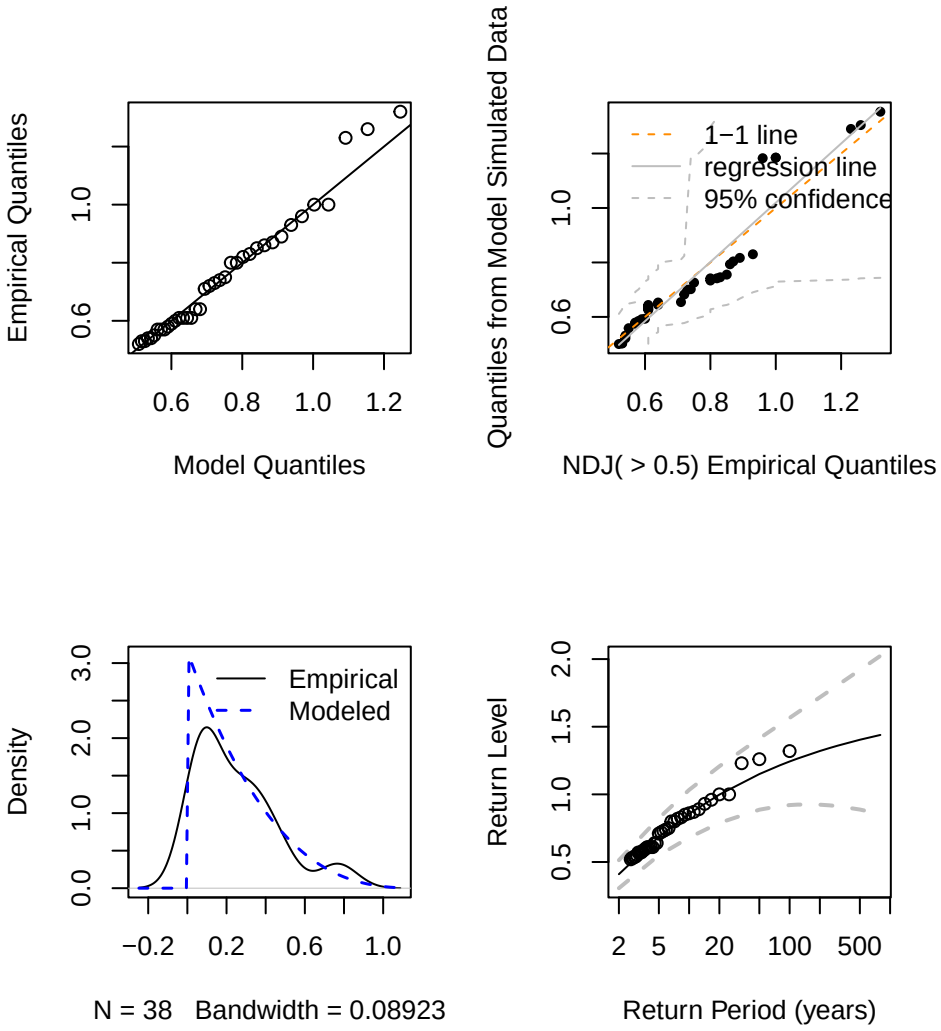
```
fit.AMJ <- fevd(AMJ, threshold=u, type="GP", time.units="91/year")
fit.NDJ <- fevd(NDJ, threshold=u, type="GP", time.units="92/year")
plot(fit.AMJ)
```

fevd(x = AMJ, threshold = u, type = "GP", time.units = "91/year")




```
plot(fit.NDJ)
```

fevd(x = NDJ, threshold = u, type = "GP", time.units = "92/year")



From the plots of subsetted data, there is visually a vast difference in the tail behaviour. From the diagnostic goodness-of-fit plots, GPD models with $u = 0.5$ seem to be reasonable models for the data in each season.
We then estimate the 2, 20 and 100-years return levels.

```
return.level(fit.AMJ, period=c(2,20,100), do.ci=T)
```

```
## fevd(x = AMJ, threshold = u, type = "GP", time.units = "91/year")
##
## [1] "Normal Approx."
##
##                         95% lower CI Estimate 95% upper CI
## 2-year return level         1.364831 1.478252     1.591673
## 20-year return level        2.352961 2.824759     3.296557
## 100-year return level       2.889235 3.944558     4.999880
```

```
return.level(fit.NDJ, period=c(2,20,100), do.ci=T)
```

```
## fevd(x = NDJ, threshold = u, type = "GP", time.units = "92/year")
##
## [1] "Normal Approx."
##
##                        95% lower CI  Estimate 95% upper CI
## 2-year return level       0.3064912 0.4099456    0.5133999
## 20-year return level      0.7892227 0.9982445    1.2072662
## 100-year return level     0.9207226 1.2429599    1.5651972
```

Not surprisingly the return level estimates are very different in the two seasons.

d) Alternatively, consider and fit within-year time-varying GPD parameter models for $\tilde{\sigma}(t)$ and $\xi(t)$. Use the possible predictors

$$\cos\left(2\pi \times \text{doy}_t/365.25\right), \qquad \sin\left(2\pi \times \text{doy}_t/365.25\right),$$

where doy_t is the day of year for observation x_t.

Is there any evidence (perform likelihood-ratio tests) of within-year variability in the GPD parameter estimates? If so, superimpose the time varying predictors on a plot of the data against doy_t.

Solution:

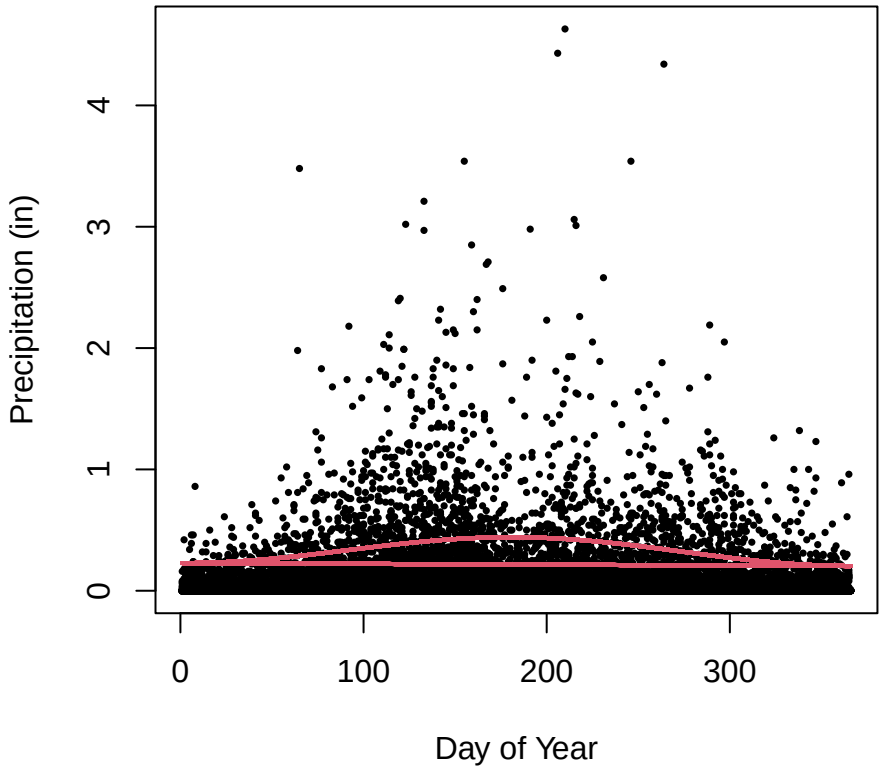
```
X <- data.frame(ccos=cos(2*pi*Fort$tobs/365.25),
                ssin=(2*pi*Fort$tobs/365.25))
fit.1 <- fevd(Fort$Prec, data=X, threshold=u, scale.fun=~ccos+ssin, type="GP")
lr.test(fit, fit.1) # Supported
```

```
##
##  Likelihood-ratio Test
##
## data:  Fort$PrecFort$Prec
## Likelihood-ratio = 17.853, chi-square critical value = 5.9915, alpha =
## 0.0500, Degrees of Freedom = 2.0000, p-value = 0.0001328
## alternative hypothesis: greater
```

```
fit.2 <- fevd(Fort$Prec, data=X, threshold=u, scale.fun=~ccos+ssin,
              shape.fun=~ccos+ssin,type="GP")
lr.test(fit.1, fit.2) # not Supported
```

```
##
##  Likelihood-ratio Test
##
## data:  Fort$PrecFort$Prec
## Likelihood-ratio = 1.7878, chi-square critical value = 5.9915, alpha =
## 0.0500, Degrees of Freedom = 2.0000, p-value = 0.4091
## alternative hypothesis: greater
```

```
p <- fit.1$results$par
plot(Fort$tobs, Fort$Prec, pch=16, cex=0.5, xlab="Day of Year",
     ylab="Precipitation (in)")
lines(Fort$tobs, (p[1]+p[2]*X$ccos+p[3]*X$ssin), col=2, lwd=2)
```



A time varying predictor for the scale parameter $\sigma(t)$ is supported with strong evidence (p-value=0.0001328), although there is little evidence that $\xi(t)$ additionally varies in this way (p-value=0.4091). Looking at the plot it is clear that an increase in scale in the middle of the year makes sense. There is no evidence for a change in the shape of the extremes over the course of the year.

e) Using the model accounting for within-year seasonality, with time-varying parameters $\tilde{\sigma}(t)$ and $\xi(t)$:

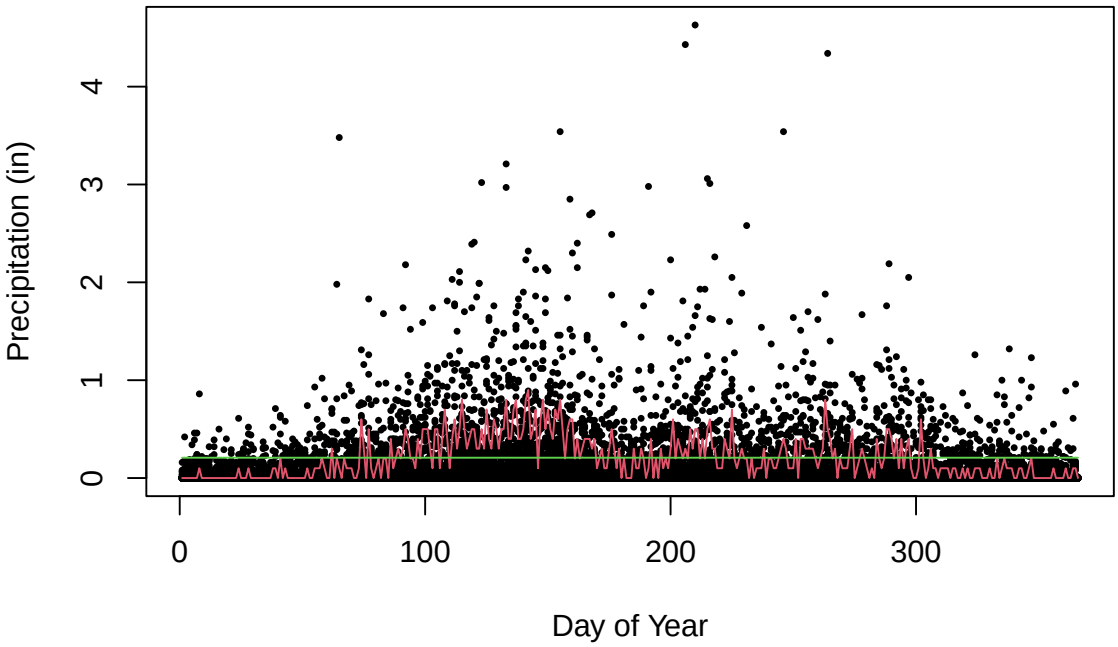
   i) Read the below information on how to estimate return levels for time-varying GPD models.

   ii) Estimate the empirical threshold exceedance rate $\eta_u(c_t)$ for each $t = 1, \ldots, 366$. Draw this rate (multiplied by 10, say) on a plot of precipitation versus day of year, and compare to the constant threshold. Is the threshold exceedance rate constant?

   Solution:

```
ind <- Fort$Prec>u
eta <- rep(NA,366)
for(i in 1:366) {
   temp = ind[Fort$tobs==i]
```

```
   eta[i] = sum(temp)/length(temp)
}
plot(Fort$tobs, Fort$Prec, pch=16, cex=0.5, xlab="Day of Year",
     ylab="Precipitation (in)")
lines(eta*10, col=2)
eta.const <- sum(Fort$Prec>u)/length(Fort$Prec)
lines(c(1,366), 10*rep(eta.const,2), col=3)
```



There are a some very clear increases in the threshold exceedance rate, e.g. around 150 and 210 days. So assuming a constant threshold here would overestimate the exceedance probability in the beginning and end of the year, and underestimate it elsewhere. It's possible that smoothing the rate would provide stability benefits in the following analysis, but we just use the empirical rate for what follows.

iii) By writing and minimising an appropriate function (see below information), estimate the 2-year, 20-year and 100-year return levels.

Solution:

```
v <- make.qcov(fit.1, vals=list(sigma1=X$ccos[1:366],
               sigma2=X$ssin[1:366])) # covariates needed for each RL

rl.solve <- function(z, m) { # m represent the return period
  pp <- pextRemes(fit.1,rep(z,366), threshold=u, lower.tail=F, qcov=v)
  abs( 1/(m*ny)-mean(pp*eta) )
}

ny <- 365.25
for(rp in c(2,20,100)){
```

```
    rl <- optim(1, rl.solve, m=rp)
    cat(rp, "-years return level: ", rl$par, "\n", sep="")
}
```

```
## 2-years return level: 1.774988
## 20-years return level: 3.405247
## 100-years return level: 4.951062
```

iv) How do these return periods compare to the return levels assuming constant GPD parameters and the empirical quantiles? Which might we prefer?

Solution:
As noisy proxies of the "truth", the empirical 2-year, 20-year and 100-year quantiles are given by:

```
quantile(Fort$Prec,probs=1-c(0.005,0.0005,0.0001))
```

```
##    99.5%   99.95%   99.99%
## 1.060000 2.397385 3.540000
```

For each set of m-year return levels we have:

$$\text{Constant GPD} > \text{Time-varying GPD} > \text{Empirical},$$

so from this (3-point!!) perspective the time-varying GPD is a better fit that the time constant GPD.

Note on estimating Time Varying Return Levels:

From the lecture notes (Week 4, Slide 33-34) we know that the definition of return level z_m and return period in years, m, is given by

$$\frac{1}{mn_y} = [1 - \mathbb{P}(X \leq z_m|\tilde{\sigma}, \xi)]\,\eta_u, \tag{1}$$

where $\eta_u = \mathbb{P}(X > u)$ is the probability that $X > u$, n_y is the number of observations in one year, and $\mathbb{P}(X \leq z_m|\tilde{\sigma}, \xi)$ is the GPD distribution function.

Now suppose that over the course of the n_y observations in one year, the upper tail of X varies according to time-based covariates, so that we have $X_1, \ldots, X_{n_y} = \{X_t\}$ with X_t depending on c_t (so that e.g.~the parameters of this distribution depend on the covariates c_t). The RHS of (1) is $\mathbb{P}(X > u + z_m) = \mathbb{P}(X > u + z_m|X > u)\mathbb{P}(X > u)$, where the first term is the GPD tail probability, and the second is the threshold exceedance rate. Letting $\mathbb{P}(X > u + z_m)$ vary over time, with X_t

depending on covariates c_t, we wish to solve

$$\begin{aligned}
\frac{1}{m} &= \mathbb{P}(X > u + z_m) \\
&= \int_{c_t} \mathbb{P}(X_t > u + z_m | C_t = c_t) f_{C_t}(c_t) dc_t \qquad (2) \\
&\approx \frac{1}{n_y} \sum_{t=1}^{n_y} \mathbb{P}(X_t > u + z_m | C_t = c_t) \qquad (3) \\
&= \frac{1}{n_y} \sum_{t=1}^{n_y} \mathbb{P}(X_t > u + z_m | C_t = c_t, X_t > u) \mathbb{P}(X_t > u | C_t = c_t) \\
&= \frac{1}{n_y} \sum_{t=1}^{n_y} \mathbb{P}(X_t > u + z_m | C_t = c_t, X_t > u) \eta_u(c_t). \qquad (4)
\end{aligned}$$

(2) comes from the law of total probability, (3) is approximately true if the observed covariates form a representative sample from $f_{C_t}(c_t)$ over the course of the year (e.g.~such as being observed uniformly over the year, and equally many times in the sample). The first term in the sum in (4) is simply a GPD probability with parameters $\tilde{\sigma}(t)$ and $\xi(t)$ as a function of c_t (time and other covariates). $\nu_u(c_t)$ is the probability that the threshold is exceeded at time point t. (So e.g.~if $t = 1$, we might estimate $\eta_u(c_1)$ as the proportion of observed data points on 1st January that exceed u, or some smoothed version of this. Alternatively we could model $\eta_u(c_t)$ parametrically over time.) Finally, as (4) defines an averaged *per-observation* relationship between return period m and return level z_m, we can adjust this to an annual (block) scale by replacing m with mn_y. That is, given m to estimate z_m we need to numerically solve (e.g. using `R` function `optim`):

$$\frac{1}{mn_y} = \frac{1}{n_y} \sum_{t=1}^{n_y} \mathbb{P}(X_t > u + z_m | C_t = c_t, X_t > u) \eta_u(c_t).$$

For further description and discussion of this approach, see e.g.~Eastoe E.G.~and J.A.~Tawn (2009). *Journal of the Royal Statistical Society, Series C*, **58**, 25-45. Available at: `https://www.jstor.org/stable/25578145`